

FIG. 1

FIELD NAME	SIZE (BITS)
PLAY LIST HEADER	8
NUMBER OF ITEMS (NOI)	8
LIST ID	16
PREVIOUS LIST OFFSET	16
NEXT LIST OFFSET	16
RETURN LIST OFFSET	16
PLAYING TIME	16
PLAY ITEM WAIT TIME	8
AUTO PAUSE WAIT TIME	8
PLAY ITEM #1 NUMBER	16
—	
—	
PLAY ITEM #NOI NUMBER	16

FIG. 2

FIELD NAME	SIZE (BITS)
SELECTION LIST HEADER	8
FLAGS	8
NUMBER OF SELECTIONS (NOS)	8
BASE OF SELECTION NUMBER (BSN)	8
LIST ID	16
PREVIOUS LIST OFFSET	16
NEXT LIST OFFSET	16
RETURN LIST OFFSET	16
DEFAULT LIST OFFSET	16
TIME-OUT LIST OFFSET	16
WAIT TIME FOR TIME-OUT	8
LOOP COUNT & JUMP TIMING	8
PLAY ITEM NUMBER	16
SELECTION #BSN OFFSET	16
—	
—	
SELECTION #(BSN + NOS-1) OFFSET	16

FIG. 3

4/6

FIELD NAME	SIZE (BITS)
COMMAND LIST HEADER	8
INSTRUCTION	40
NEXT_LIST OFFSET	16

FIG. 4

BYTE #1 0..2 3..4 5..7			BYTE #2	BYTE #3	BYTE #4	BYTE #5	ABBREVIATION AND DESCRIPTION	
001	cond0	calc	i	j	k	l	calc	IF cond0 { V [j] := V [k] opcode V L }
010	cond0	000	i	--	k	l	move0	IF cond0 { V [k] := V L }
	cond1	001	i	j	k	l	move1	IF cond1 { V [k] := V L }
	cond0	010	i	j	dd	dd	fill0	IF cond0 { V [j] := dddd }
	00	110	i	j	dd	dd	fillr	WHILE V [i] >= 0 { V [j] + V [i] := dddd; V [i] -- }
100	cond0	000	i	--	offs	offs	jump0	IF cond0 { goto offs }
	cond1	001	i	j	offs	offs	jump1	IF cond1 { goto offs }
	cond0	010	i	j	--	--	return	IF cond0 { goto V [j] }
	cond0	100	i	--	offs	offs	loop0	IF cond0 { dec (V [i]; goto offs }
	cond1	101	i	j	offs	offs	loop1	IF cond1 { dec (V [i]; goto offs }
	cond0	110	i	j	offs	offs	loop2	IF cond0 { dec (V [i], V [j]; goto offs }
110	cond0	000	i	j	offs	offs	jumpw	IF cond0 { wait V [j] SECONDS FOR INPUT; goto offs }

FIG. 5

5/6

BIT 3..4	MEANING
% 00	TRUE
% 01	$V[i] > 0$
% 10	$V[i] < 0$
% 11	$V[i] = 0$

FIG. 6

BIT 3..4	MEANING
% 00	TRUE
% 01	$V[i] > V[j]$
% 10	$V[i] < V[j]$
% 11	$V[i] = V[j]$

FIG. 7

BIT 5..7	MEANING
% 000	ADD
% 001	MIN
% 010	MUL
% 011	DIV
% 100	MOD
% 101	AND
% 110	OR
% 111	XOR

FIG. 8

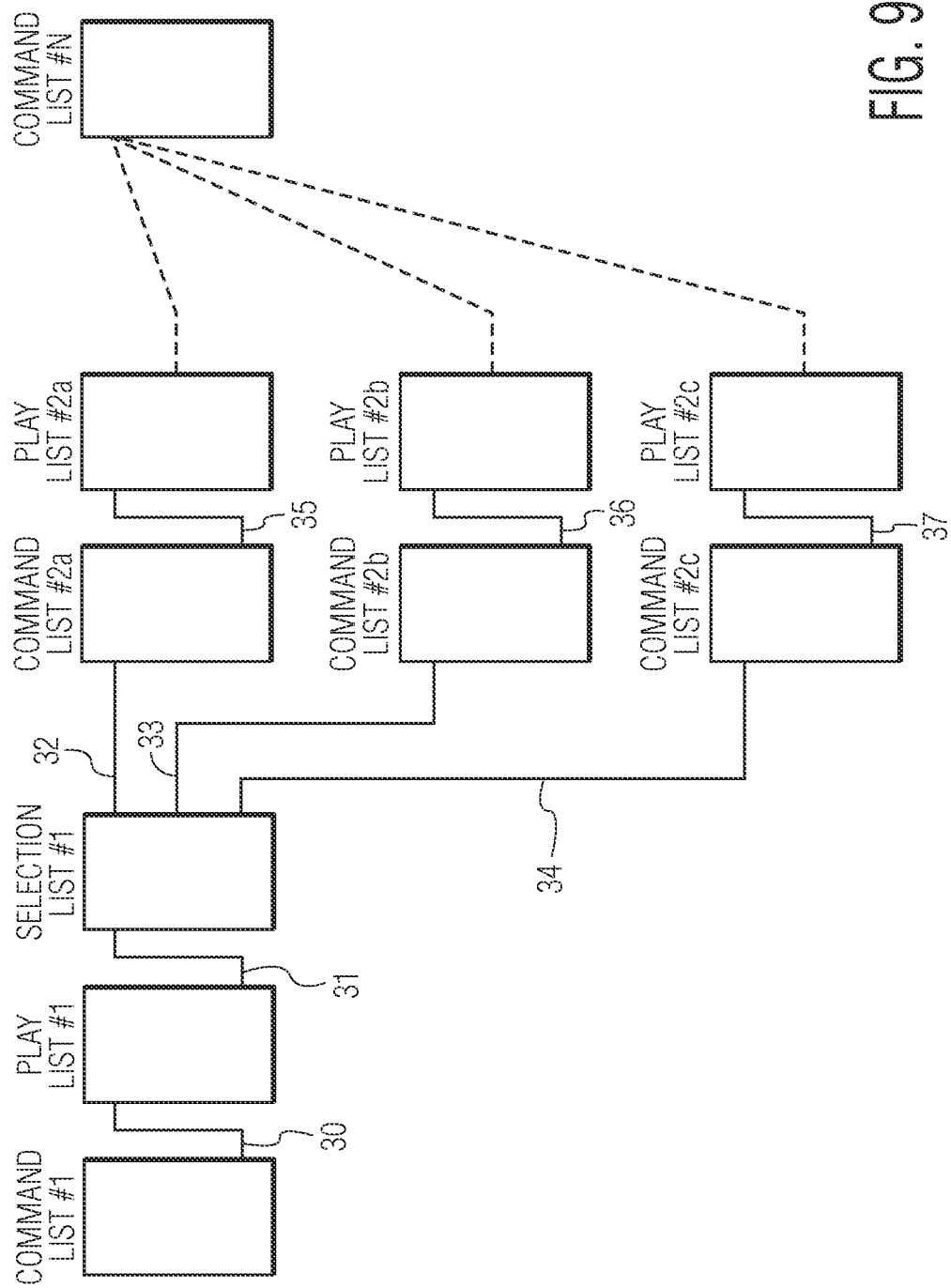


FIG. 9